Strand Spaces: Why is a Security Protocol Correct?*

F. Javier Thayer Fábrega Jonathan C. Herzog
Joshua D. Guttman
The MITRE Corporation
{jt, jherzog, guttman}@mitre.org

Abstract

A strand is a sequence of events; it represents either the execution of legitimate party in a security protocol or else a sequence of actions by a penetrator. A strand space is a collection of strands, equipped with a graph structure generated by causal interaction. In this framework, protocol correctness claims may be expressed in terms of the connections between strands of different kinds.

In this paper we develop the notion of a strand space. We then prove a generally useful lemma, as a sample result giving a general bound on the abilities of the penetrator in any protocol. We apply the strand space formalism to prove the correctness of the Needham-Schroeder-Lowe protocol. Our approach gives a detailed view of the conditions under which the protocol achieves authentication and protects the secrecy of the values exchanged. We also use our proof methods to explain why the original Needham-Schroeder protocol fails.

We believe that our approach is distinguished from other work on protocol verification by the simplicity of the model and the ease of producing intelligible and reliable proofs of protocol correctness even without automated support.

1 Introduction

A security protocol is a sequence of messages between two or more parties in which encryption is used to provide authentication or to distribute cryptographic keys for new conversations [17]. Even when security protocols have been developed carefully by experts and reviewed carefully by other experts, they are often found later to have flaws that make them unusable (see, for example, [6, 11]). In many cases, the attacks do not presuppose any weakness in the cryptosystem being used, and would be just as harmful with an ideal cryptosystem. In other cases, characteristics of the cryptosystem and characteristics of the protocol combine to cause protocol failure [16, 5, 18].

Analyzing security protocols consists mainly in two complementary activities. The first is to find flaws in those protocols that are not correct, and the second is to establish convincingly the correctness of those that are. These activities are interrelated, because the discovery of a flaw may suggest an altered protocol that we may wish to prove correct, and because a failure to prove the correctness of a protocol may suggest a particular flaw.

In this paper, however, we focus on the second activity, proving the correctness of protocols when they are in fact correct. Moreover, at this stage, we consider only protocol correctness assuming ideal cryptography.

Much work both recently (for instance, [1, 21, 24]) and of an earlier vintage (such as [7, 3]) has proposed techniques for proving protocols correct. We believe that the approach presented here has several advantages. First, our approach gives a clear semantics to the assumption that certain data items, such as nonces and session keys, are fresh, and never arise in more than one protocol run. Second, our approach works with an explicit model of the possible behaviors of a system penetrator; this allows us to develop general theorems that bound the abilities of the penetrator, independent of the protocol under study. One such theorem is presented below in Section 3.2. Third, our approach allows various notions of correctness, involving both secrecy and authentication, to be stated and proved. And finally, in our opinion, the approach leads to detailed insight into the reasons why the protocol is correct, and the assumptions required. Proofs are simple and informative: they are easily developed by hand, and they help to identify more exact conditions under which we can rely on the protocol.

Our basic contribution is the *strand space*. A *strand* is a sequence of events that a participant may engage in. For a legitimate participant, each strand is a sequence of message sends and receives; it represents the actions of that party (but of that party only, not its presumed interlocutor) in a particular run of the protocol, with specific values of all data

*This work was supported by the National Security Agency through US Army CECOM contract DAAB 07-96-C-E601. Published in *Proceedings, 1998 IEEE Symposium on Security and Privacy*, 3-6 May 1998 in Oakland, California.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
1998		**00-00-1998 to 00-00-1998**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Strand Spaces: Why is a Security Protocol Correct?	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **MITRE Corporation,202 Burlington Road,Bedford,MA,01730-1420**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**12**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

items such as keys and nonces. A strand for a penetrator is a sequence of message sends and receives possible for the penetrator. Penetrator strands include such activities as:

- receiving a symmetric key and a message encrypted using that key, and then sending the result of decrypting the message;
- receiving two messages and sending the result of concatenating them;
- sending out a guessable data item such as a name; and so on.

A *strand space* is a set of strands consisting of strands for the various legitimate protocol parties, together with penetrator strands.

A *bundle* consists of a number of strands—legitimate or otherwise—hooked together where one strand sends a message and another strand receives that same message. Typically, for a protocol to be *correct*, each such bundle must consist of one strand for each of the legitimate parties, all agreeing on the participants, nonces, and session keys [14, 23, 27]. Penetrator strands may also be entangled in a bundle, even in a correct protocol, but they do not prevent the legitimate parties from agreeing on the data values, or from maintaining the secrecy of the values chosen.

Protocol correctness typically depends essentially on the *freshness* of data items such as nonces and session keys. For this reason, the strand spaces that concern us are not full, in the sense that they do not contain all the strands that would arise if every participant used every possible data item. A strand space models the fact that some values occur only freshly by including only one strand *originating* that data item by initially sending a message containing it. Many strands, by contrast, may stand ready to combine with the originating strand by receiving the message and processing its contents further. A strand space will also model the assumption that some values are impossible for a penetrator to guess; in essence, the space simply lacks any penetrator strand in which this value is sent without having first been received.

In this paper, we will develop the basic machinery of strand spaces (Section 2). This machinery includes a partial order that models causal contribution, and justifies an induction-like proof method (Section 2.2). We then develop our model of the penetrator (Section 3), including a simple but useful theorem that gives a general bound on what the penetrator can do, regardless of the protocol being modeled (Section 3.2). In Section 4, we study the Needham-Schroeder-Lowe public key protocol [17, 11, 12] as an example, proving both an authentication result (Section 4.2) and a secrecy result (Section 4.4).

A technical report [25] develops more powerful bounds on the penetrator, akin to the one in Section 3.2. These are then used to prove authentication and secrecy results for two other protocols, namely the Otway-Rees protocol and the Yahalom protocols. In each case, we discover detailed (and unexpected) information on the exact conditions under which the protocol is correct.

2 Strand Spaces

In this section, we will introduce strand spaces and related notions (Section 2.1). A *bundle* is a portion of a strand space large enough to represent a full protocol exchange; it has a natural causal precedence relation relative to which inductive arguments may be carried out (Section 2.2). The terms that we will consider in the present paper are described in Section 2.3; a less restrictive treatment is available in [25], but would merely distract from the main points here. We finish this section by summarizing some of the notions of correctness that are natural to state and prove in our context (Section 2.4).

2.1 Basic Notions

Consider a set $\mathbf{A}$, the elements of which are the possible messages that can be exchanged between principals in a protocol. We will refer to the elements of $\mathbf{A}$ as *terms*. In the applications that we consider, the set $\mathbf{A}$ has more structure, but in this section we assume that at least a *subterm* relation is defined on $\mathbf{A}$. $t_1 \sqsubset t$ means t_1 is a subterm of t. In a protocol, principals can either send or receive terms. We will represent sending a term as the occurrence of that term with positive sign, and receiving a term as its occurrence with a negative sign.

Definition 2.1 *A* signed term *is a pair $\langle \sigma, a \rangle$ with $a \in \mathbf{A}$ and σ one of the symbols $+, -$. We will write a signed term as $+t$ or $-t$. $(\pm\mathbf{A})^*$ is the set of finite sequences of signed terms. We will denote a typical element of $(\pm\mathbf{A})^*$ by $\langle \langle \sigma_1, a_1 \rangle, \ldots, \langle \sigma_n, a_n \rangle \rangle$.*

By abuse of language, we will still treat signed terms as ordinary terms, for instance as having subterms.

Definition 2.2 *A* strand space *is a set Σ with a* trace *mapping $tr : \Sigma \to (\pm\mathbf{A})^*$.*

In particular applications of the theory, the mapping tr need not be injective, because we may want to distinguish between various instances of the same trace. For instance, to model authentication properties of certain protocols it may be necessary to distinguish identical traces originating from different principals, or to model simple replay attacks we may need to distinguish identical traces originating successively from the same principal.

Fix a strand space Σ.

1. A *node* is a pair $\langle s, i \rangle$, with $s \in \Sigma$ and i an integer satisying $1 \leq i \leq \text{length}(\text{tr}(s))$. The set of nodes is denoted by $\mathcal{N}$. We will say the node $\langle s, i \rangle$ belongs to the strand s. Clearly, every node belongs to a unique strand.

2. If $n = \langle s, i \rangle \in \mathcal{N}$ then $\text{index}(n) = i$ and $\text{strand}(n) = s$. Define $\text{term}(n)$ to be $(\text{tr}(s))_i$, i.e. the ith signed term in the trace of s. Similarly, $\text{uns\_term}(n)$ is $((\text{tr}(s))_i)_2$, i.e. the unsigned part of the ith signed term in the trace of s.

3. If $n_1, n_2 \in \mathcal{N}$, $n_1 \rightarrow n_2$ means $\text{term}(n_1) = +a$ and $\text{term}(n_2) = -a$. It means that node n_1 sends the message a, which is received by n_2, creating a causal link between their strands.

4. If $n_1, n_2 \in \mathcal{N}$, then $n_1 \Rightarrow n_2$ means n_1, n_2 occur on the same strand with $\text{index}(n_1) = \text{index}(n_2) - 1$. It expresses that n_1 is an immediate causal predecessor of n_2 on the strand.

5. An unsigned term t *occurs in* $n \in \mathcal{N}$ iff $t \sqsubset \text{term}(n)$.

6. An unsigned term t *originates* on $n \in \mathcal{N}$ iff: $\text{term}(n)$ is positive; $t \sqsubset \text{term}(n)$; and whenever n' precedes n on the same strand, $t \not\sqsubset \text{term}(n')$.

7. An unsigned term t is *uniquely originating* iff t originates on a unique $n \in \mathcal{N}$.

If a term t originates uniquely in a particular strand space, then it can play the role of a nonce or session key in that structure.

$\mathcal{N}$ becomes an ordered graph with both sets of edges $n_1 \rightarrow n_2$ and $n_1 \Rightarrow n_2$.

2.2 Bundles and Causal Precedence

A *bundle* is a finite subgraph of this graph, for which we can regard the edges as expressing the causal dependencies of the nodes.

Definition 2.3 *Let $\mathcal{C}$ be a set of edges, and let $\mathcal{N}_\mathcal{C}$ be the set of nodes incident with any edge in $\mathcal{C}$. $\mathcal{C}$ is a bundle if:*

1. *$\mathcal{C}$ is finite.*
2. *If $n_1 \in \mathcal{N}_\mathcal{C}$ and $\text{term}(n_1)$ is negative, then there is a unique n_2 such that $n_2 \rightarrow n_1 \in \mathcal{C}$.*
3. *If $n_1 \in \mathcal{N}_\mathcal{C}$ and $n_2 \Rightarrow n_1$ then $n_2 \Rightarrow n_1 \in \mathcal{C}$.*
4. *$\mathcal{C}$ is acyclic.*

Notational Convention 2.4 *A node n is in a bundle $\mathcal{C}$, written $n \in \mathcal{C}$, if $n \in \mathcal{N}_\mathcal{C}$; a strand s is in a bundle if all of its nodes are in $\mathcal{N}_\mathcal{C}$.*

Definition 2.5 *Suppose that $\mathcal{S}$ is a set of edges, i.e. a subset of the union of $\rightarrow$ and $\Rightarrow$, and let $\mathcal{N}_\mathcal{S}$ be the set of nodes incident with any edge in $\mathcal{S}$.*

Then $\prec_\mathcal{S}$ is the transitive closure of $\mathcal{S}$, and $\preceq_\mathcal{S}$ is the reflexive, transitive closure of $\mathcal{S}$; each is a subset of $\mathcal{N}_\mathcal{S} \times \mathcal{N}_\mathcal{S}$.

$n \prec_\mathcal{S} n'$ means that there is a sequence of one or more edges (of either kind) belonging to $\mathcal{S}$ leading from n to n'. Similarly, $n \preceq_\mathcal{S} n'$ means that there is a sequence of zero or more edges belonging to $\mathcal{S}$ leading from n to n'. In case $\mathcal{S}$ is a bundle, $\preceq_\mathcal{S}$ is a partial ordering. We regard it as expressing causal precedence, because $n \prec_\mathcal{S} n'$ holds just in case n's occurrence contributes to allowing n' to occur.

Lemma 2.6 *Suppose $\mathcal{C}$ is a bundle. Then $\preceq_\mathcal{C}$ is a partial order, i.e. a reflexive, antisymmetric, transitive relation. Every non-empty subset of the nodes in $\mathcal{C}$ has $\preceq_\mathcal{C}$-minimal members.*

When a bundle $\mathcal{C}$ is understood, we will simply write $\preceq$. Similarly, "minimal" will mean $\preceq_\mathcal{C}$-minimal.

Most of our arguments turn on the $\preceq_\mathcal{C}$-minimal elements in some set of nodes. These arguments are motivated by the question, "What did he know, and when did he know it?" The existence of minimal members in non-empty sets serves as a kind of induction principle, an observation that clarifies the relation of our approach to Paulson's and Schneider's [21, 24].

Lemma 2.7 *Suppose $\mathcal{C}$ is a bundle, and suppose S is a set of nodes such that $\text{uns\_term}(m) = \text{uns\_term}(m')$ implies that $m \in S$ iff $m' \in S$, for all nodes m, m'. If n is a $\preceq_\mathcal{C}$-minimal member of S, then the sign of n is positive.*

PROOF. If $\text{term}(n)$ were negative, then by the bundle property, $n' \rightarrow n$ for some $n' \in \mathcal{C}$ and sign apart, $\text{term}(n) = \text{term}(n')$. Hence, $n' \in S$, violating the minimality property of n. ■

Lemma 2.8 *Suppose $\mathcal{C}$ is a bundle, $t \in \mathbf{A}$ and $n \in \mathcal{C}$ is a $\preceq_\mathcal{C}$-minimal element of $\{m \in \mathcal{C} : t \sqsubset \text{term}(m)\}$. The node n is an originating occurrence for t.*

PROOF. By Lemma 2.7, the sign of n is positive. If $n' \prec n$ lies on the strand of n, then $n' \in \mathcal{C}$, so by the minimality property of n, $t \not\sqsubset \text{term}(n')$. Thus n is originating for t. ■

2.3 Terms and Encryption

We will now specialize the set of terms $\mathbf{A}$. In particular we will assume given:

- A set $\mathbf{T}$ of texts (representing the atomic messages).

- A set $\mathbf{K}$ of cryptographic keys disjoint from $\mathbf{T}$, equipped with a unary operator inv : $\mathbf{K} \to \mathbf{K}$. We assume that inv maps each member of a key pair for an asymmetric cryptosystem to the other, and that it maps a symmetric key to itself.

- Two binary operators

$$\text{encr} : \mathbf{K} \times \mathbf{A} \to \mathbf{A}$$
$$\text{join} : \mathbf{A} \times \mathbf{A} \to \mathbf{A}$$

As usual, we will write $\text{inv}(K)$ as K^{-1}, $\text{encr}(K, m)$ as $\{m\}_K$, and $\text{join}(a, b)$ as $a\, b$.

The proofs in this paper will use an assumption we will call the assumption of free encryption; many other authors (e.g. [13, 15, 21]) make similar assumptions. It stipulates that a ciphertext can be regarded as a ciphertext in just one way:

$$\{m\}_K = \{m'\}_{K'} \Longrightarrow m = m' \wedge K = K'$$

For the purposes of this paper we will make a stronger assumption, namely that $\mathbf{A}$ is the algebra freely generated from $\mathbf{T}$ and $\mathbf{K}$ by the two operators encr and join, in the sense that these two operators are injective, and have range disjoint from each other and from $\mathbf{T}$ and $\mathbf{K}$. This is more than would be needed for our method [25], but it leads to the simplest exposition of the main points.

Attacks that might exist if there are terms that may be "read" as having more than one form are referred to as *type flaw attacks* [4]. Some type flaw attacks seem implausible, in the sense that most implementations would not be vulnerable to them, while others are more troublesome. Type flaws could be modeled by extending strand spaces in various possible ways.

The subterm relation $\sqsubset$ is defined inductively, so that:

- $a \sqsubset t$ for $t \in \mathbf{T}$ iff $a = t$;
- $a \sqsubset K$ for $K \in \mathbf{K}$ iff $a = K$;
- $a \sqsubset \{g\}_K$ iff $a \sqsubset g$ or $a = \{g\}_K$;
- $a \sqsubset g\, h$ iff $a \sqsubset g$, $a \sqsubset h$ or $a = g\, h$.

We should emphasize that, for $K \in \mathbf{K}$, $K \sqsubset \{g\}_K$ only if $K \sqsubset g$ already. Restricting subterms in this way reflects an assumption about the penetrator's capabilities, to wit, that keys can be obtained from cyphertext only if they are embedded in the text that was encrypted. This might not always be the case—for instance, if a dictionary attack is possible—but it is the assumption we will make here.

This notion of subterm does not always mesh perfectly with the definition of origination and unique origination, which refers to the subterm relation (Section 2.1, Clauses 6 and 7). In some cases [26], it is more natural to use a notion of origination referring to the larger relation $\sqsubset'$; that relation would be defined so that

$$a \sqsubset' \{g\}_K \quad \text{iff} \quad a \sqsubset' g \ \vee \ a = K \ \vee \ a = \{g\}_K$$

2.4 Notions of Correctness

Gavin Lowe studies a range of authentication properties in [14]; strand spaces are a natural model for stating and proving his *agreement* properties.[1] A protocol guarantees a participant B (say, as the responder) agreement for certain data items $\vec{x}$ if:

> each time a principal B completes a run of the protocol as responder using $\vec{x}$, apparently with A, then there is a unique run of the protocol with the principal A as initiator using $\vec{x}$, apparently with B.

A weaker non-injective agreement does not ensure uniqueness, but requires only:

> each time a principal B completes a run of the protocol as responder using $\vec{x}$, apparently with A, then there exists a run of the protocol with the principal A as initiator using $\vec{x}$, apparently with B.

Non-injective agreement is weaker because it does not prevent the other party A from being duped into executing multiple runs matching a single run by B.

We can prove non-injective agreement by establishing that, whenever a bundle $\mathcal{C}$ contains a responder strand using $\vec{x}$, then $\mathcal{C}$ also contains an initiator strand using $\vec{x}$. We can establish agreement by showing that $\mathcal{C}$ contains a *unique* initiator strand using $\vec{x}$. We will illustrate these properties in Propositions 4.2 and 4.8.

A simple notion of secrecy, sufficient for our purposes here, for a data value x may also be easily stated. We stipulate that no node n—whether a regular node or a penetrator node—ever has x unprotected as its term. Thus, a value x is secret in a strand space Σ if, for every bundle $\mathcal{C}$ in Σ, and every node $n \in \mathcal{C}$, $\text{term}(n) \neq x$. We illustrate this property in Proposition 4.10.

This notion of secrecy concerns only what is "said on the wire." In this sense, a value is secret if the non-penetrator strands never emit it, and the penetrator can never derive (and emit) it from what they do emit. Legitimate protocol participants may "know" a secret value in the sense of carrying out computations that depend on it, so long as their behavior in the protocol does not include disclosing it in public.

[1] These are akin to the *correspondence* properties of Woo and Lam [27].

More stringent notions of secrecy are also possible, as for instance the information flow security properties, and may be fruitfully applied to security protocols [8].

3 The Penetrator

The penetrator's powers are characterized by two ingredients, namely a set of keys known initially to the penetrator and a set of penetrator strands that allow the penetrator to generate new messages from messages he intercepts.

A *penetrator set* consists of a set of keys $\mathbf{K}_{\mathcal{P}}$. It contains the keys initially known to the penetrator. Typically it would contain: all public keys; all private keys of penetrators; and all symmetric keys K_{px}, K_{xp} initially shared between the penetrator and principals playing by the protocol rules. It may also contain "lost keys" that became known to the penetrator, either because a principal was careless, or else because the penetrator succeeded in some cryptanalysis.

3.1 Penetrator Strands

The atomic actions available to the penetrator are encoded in a set of *penetrator traces*. They summarize his ability to discard messages, generate well known messages, piece messages together, and apply cryptographic operations using keys that become available to him. A protocol attack typically requires hooking together several of these atomic actions.

Definition 3.1 *A* penetrator trace *is one of the following:*

M. *Text message:* $\langle +t \rangle$ *where* $t \in \mathbf{T}$

F. *Flushing:* $\langle -g \rangle$

T. *Tee:* $\langle -g,\ +g,\ +g \rangle$

C. *Concatenation:* $\langle -g,\ -h,\ +g\, h \rangle$

S. *Separation into components:* $\langle -g\, h,\ +g,\ +h \rangle$

K. *Key:* $\langle +K \rangle$ *where* $K \in \mathbf{K}_{\mathcal{P}}$.

E. *Encryption:* $\langle -K,\ -h,\ +\{h\}_K \rangle$.

D. *Decryption:* $\langle -K^{-1},\ -\{h\}_K,\ +h \rangle$.

This set of penetrator traces gives the penetrator powers similar to those in other approaches, e.g. [13, 21]. They ensure that the values that may be emitted by the penetrator are closed under joining, encryption, and the relevant "inverses."

It is also possible to extend the set of penetrator traces given here if it is desired to model some special ability of the penetrator. That requires no essential change to our overall framework, although the proofs in this paper would then need to be modified to take account of the additional penetrator traces. Our theorems characterize a penetrator with just the powers we have described; a penetrator with additional computational or cryptanalytic abilities may not be subject to the same limitations.

Definition 3.2 *An* infiltrated strand space *is a pair* $(\Sigma, \mathcal{P})$ *with* Σ *a strand space and* $\mathcal{P} \subseteq \Sigma$ *such that* $tr(p)$ *is a penetrator trace for all* $p \in \mathcal{P}$.

A strand $s \in \Sigma$ *is a* penetrator strand *if it belongs to* $\mathcal{P}$*, and a node is a* penetrator node *if the strand it lies on is a penetrator strand. Otherwise we will call it a* non-penetrator *or* regular *strand or node.*

A node n *is a* **M***,* **F***, etc. node if* n *lies on a penetrator strand with a trace of kind* **M***,* **F***, etc.*

We would not expect an infiltrated strand space to realize all of the penetrator traces of type **M**. In that case, the space could not model unguessable nonces. The more useful spaces Σ lack **M**-strands for many text values, which the legitimate participants can use as fresh nonces.

3.2 A Bound on the Penetrator

Because the powers of the penetrator are defined by the penetrator keys and the penetrator strands, they are independent of the choice of a particular protocol to be proved correct. We can accordingly prove general facts about the penetrator's powers, re-using them whenever we become interested in a new protocol. In [25], we develop several powerful theorems about the penetrator, which are of use in all three of the protocols studied there. Here, we will prove a simple theorem that is useful in the example we will turn to next, namely the Needham-Schroeder-Lowe protocol.

The proof of this theorem is typical of how we use Lemma 2.6. By "$S \setminus T$" we mean the set difference of S and T.

Proposition 3.3 *Let* $\mathcal{C}$ *be a bundle, and let* $K \in \mathbf{K} \setminus \mathbf{K}_{\mathcal{P}}$.

If K *never originates on a regular node, then* $K \not\sqsubset term(p)$ *for any penetrator node* $p \in \mathcal{C}$.

PROOF. Consider the set $S = \{n \in \mathcal{C} : K \sqsubset \text{term}(n)\}$. Suppose (to derive a contradiction) that S is non-empty. Then S has members that are minimal relative to $\preceq_{\mathcal{C}}$ (Lemma 2.6). By Lemma 2.8, any $\preceq_{\mathcal{C}}$-minimal members of S are originating occurrences of K. Hence, by the assumption, they are all penetrator nodes. By Lemma 2.7, they are all positive nodes. We will now examine the possible cases for positive penetrator nodes.

M. The strand has the form $\langle +t \rangle$ where $t \in \mathbf{T}$, but $K \not\sqsubset t$.

F. The strand has the form $\langle -g \rangle$, and thus lacks any positive nodes.

T. The strand has the form $\langle -g,\ +g,\ +g \rangle$, so no value originates on the positive nodes.

C. The strand has the form $\langle -g,\ -h,\ +g\, h \rangle$, so no value originates on the positive node.

S. The strand has the form $\langle -g\, h,\ +g,\ +h \rangle$, so no value originates on the positive nodes.

K. The strand has the form $\langle +K_0 \rangle$ where $K_0 \in \mathbf{K}_{\mathcal{P}}$. But $K \sqsubset K_0$ iff $K = K_0$, contrary to the assumption that $K \in \mathbf{K} \setminus \mathbf{K}_{\mathcal{P}}$.

E. The strand has the form $\langle -K_0,\ -h,\ +\{h\}_{K_0} \rangle$. By the definition of $\sqsubset$, $a \sqsubset \{h\}_{K_0}$ iff $a \sqsubset h$ or $a = \{h\}_{K_0}$. Hence, no key can occur in the positive node without having occurred in a previous node.

D. The strand has the form $\langle -K_0^{-1},\ -\{h\}_{K_0},\ +h \rangle$. By the definition of $\sqsubset$, $a \sqsubset h$ only if $a \sqsubset \{h\}_{K_0}$, so no key can occur in the positive node without having occurred in a previous node.

Hence S is in fact empty. But if S is empty, then $K \not\sqsubset \text{term}(n)$ for any $n \in \mathcal{C}$, hence certainly $K \not\sqsubset \text{term}(p)$ for penetrator nodes $p \in \mathcal{C}$. ■

This proof method is characteristic: it successively considers the minimal elements in a set, considers whether they are regular nodes or penetrator nodes, and finally takes cases on the different forms of penetrator strands.

4 The Needham-Schroeder-Lowe Protocol

This protocol was proposed by Gavin Lowe [12] as a way to fix the public-key protocol proposed by Needham and Schroeder [17], which he had discovered to be flawed [11]. In the form Lowe considers, the protocol assumes that each participant has somehow discovered the other's public key.

1. $A \longrightarrow B$: $\{N_a\ A\}_{K_B}$
2. $B \longrightarrow A$: $\{N_a\ N_b\ B\}_{K_A}$
3. $A \longrightarrow B$: $\{N_b\}_{K_B}$

The intended result of this protocol is that the two participants should come to share access to the values N_a and N_b, each associating these values with the other participant, and no other party should be in possession of them. The protocol might be used in a context where the two values are hashed together to yield a shared symmetric key for an encrypted session, for instance. This protocol differs from the original Needham-Schroeder public key protocol only in message 2; in the original protocol, B's name is not included.

In [12], Lowe proves the correctness of the revised protocol, showing that any attack against the revised protocol could be realized using just two runs of the protocol. The FDR model checker discloses that no attack exists on such a small system; this result is confirmed by examining the possible forms of an attack. In this section we will give a different proof using the strand space approach.

We specialize the term algebra somewhat, equipping it with:

- A set of names $\mathbf{T}_{\text{name}} \subseteq \mathbf{T}$. We will use variables such as A, B to range over $\mathbf{T}_{\text{name}}$.
- A mapping $K : \mathbf{T}_{\text{name}} \to \mathbf{K}$. This is the mapping that associates a public key with each principal. We will follow tradition by writing $K(A)$ in the form K_A. We will assume that this function is injective, so that if $K_A = K_B$, then $A = B$. The protocol does not achieve its authentication goals unless the mapping K is injective.

4.1 NSL Strand Spaces

Definition 4.1 *An infiltrated strand space $\Sigma, \mathcal{P}$ is an* NSL *space if Σ is the union of three kinds of strands:*

1. *Penetrator strands $s \in \mathcal{P}$;*
2. *"Initiator strands" with trace* $\text{Init}[A, B, N_a, N_b]$*, defined to be:*

   $$\langle +\{N_a\ A\}_{K_B},\quad -\{N_a\ N_b\ B\}_{K_A},\quad +\{N_b\}_{K_B} \rangle$$

   where $A, B \in \mathbf{T}_{name}$, $N_a, N_b \in \mathbf{T}$ but $N_a \notin \mathbf{T}_{name}$.
3. *Complementary "responder strands" with trace* $\text{Resp}[A, B, N_a, N_b]$*, defined to be:*

   $$\langle -\{N_a\ A\}_{K_B},\quad +\{N_a\ N_b\ B\}_{K_A},\quad -\{N_b\}_{K_B} \rangle$$

   where $A, B \in \mathbf{T}_{name}$, $N_a, N_b \in \mathbf{T}$ but $N_b \notin \mathbf{T}_{name}$.

If s is a regular strand with trace $\text{Init}[A, B, N_a, N_b]$ or $\text{Resp}[A, B, N_a, N_b]$, then we refer to A and B as the initiator and the responder of s (respectively), and to N_a and N_b as the initiator's value and responder's value (respectively). The intention is that these values should be nonces, in the sense of texts uniquely originating in Σ. Note that given any strand s in Σ, we can uniquely classify it as a penetrator strand, an initiator's strand, or a respondent's strand just by the form of its trace. In particular, given an NSL space Σ, we can read off which strands are penetrator strands, so that $(\Sigma, \mathcal{P})$ is uniquely determined. Hence we can omit $\mathcal{P}$ safely.

4.2 Agreement: The Responder's Guarantee

Proposition 4.2 *Suppose:*

1. Σ *is an NSL space and* $\mathcal{C}$ *is a bundle containing a responder's strand* s *with trace* $\text{Resp}[A, B, N_a, N_b]$*;*
2. $K_A^{-1} \notin \mathbf{K}_\mathcal{P}$*; and*
3. $N_a \neq N_b$ *and* N_b *is uniquely originating in* Σ*.*

Then $\mathcal{C}$ *contains an initiator's strand* t *with trace* $\text{Init}[A, B, N_a, N_b]$.

We will prove this using a sequence of lemmas. Throughout the remainder of this section, we will fix an arbitrary Σ, $\mathcal{C}$, s, A, B, N_a, and N_b satisfying the hypotheses of Proposition 4.2. The node $\langle s, 2\rangle$ outputs the value $\{N_a\, N_b\, B\}_{K_A}$; for convenience we will refer to this node as n_0, and to its term as v_0. The node $\langle s, 3\rangle$ receives the value $\{N_b\}_{K_B}$; we will refer to this node as n_3 and its term as v_3. We will identify two additional nodes n_1 and n_2 during the course of the proof, such that $n_0 \prec n_1 \prec n_2 \prec n_3$.

Lemma 4.3 N_b *originates at* n_0.

PROOF. By the assumptions, $N_b \sqsubset v_0$, and the sign of n_0 is positive. Thus, we need only check that $N_b \not\sqsubset n'$, where n' is the node $\langle s, 1\rangle$ preceding n_0 on the same strand. Since $\text{term}(n') = \{N_a\, A\}_{K_B}$, we need to check that $N_b \neq N_a$, which is a hypothesis, and $N_b \neq A$, which follows from the stipulation—in Definition 4.1 Clause 3—that the responder's value not be in $\mathbf{T}_{\text{name}}$. ■

Next comes the main lemma, which establishes that the crucial step is taken by a regular strand and not a penetrator strand. As usual, it considers the $\preceq$-minimal members of a set of nodes. The content of the lemma is represented in Figure 1.

Lemma 4.4 *The set* $S = \{n \in \mathcal{C} : N_b \sqsubset \mathit{term}(n) \wedge v_0 \not\sqsubset \mathit{term}(n)\}$ *has a* $\preceq$*-minimal node* n_2*. The node* n_2 *is regular, and the sign of* n_2 *is positive.*

PROOF. Because $n_3 \in \mathcal{C}$, and n_3 contains N_b but not as a subterm of v_0, S is non-empty. Hence S has (at least) a $\preceq$-minimal element n_2 by Lemma 2.6. The sign of n_2 is positive by Lemma 2.7.

Can n_2 lie on a penetrator strand p? Let us examine the possible cases for positive penetrator nodes, according to the form of the trace of p. We will consider case **S** last.

M. The trace $\text{tr}(p)$ has the form $\langle +t\rangle$ where $t \in \mathbf{T}$; so we must have $t = N_b$. In this case N_b originates on this strand. But that is impossible, as N_b originates uniquely on n_0 (Lemma 4.3).

F. The trace $\text{tr}(p)$ has the form $\langle -g\rangle$, and thus lacks any positive nodes.

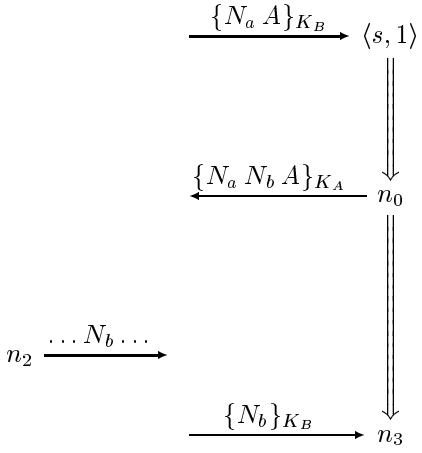

Figure 1. Regular Node n_2: Minimal in S

T. The trace $\text{tr}(p)$ has the form $\langle -g,\ +g,\ +g\rangle$, so the positive nodes are not minimal occurrences.

C. The trace $\text{tr}(p)$ has the form $\langle -g,\ -h,\ +g\, h\rangle$, so the positive node is not a minimal occurrence.

K. The trace $\text{tr}(p)$ has the form $\langle +K_0\rangle$ where $K_0 \in \mathbf{K}_\mathcal{P}$. But $N_b \not\sqsubset K_0$, so this case does not apply.

E. The trace $\text{tr}(p)$ has the form $\langle -K_0,\ -h,\ +\{h\}_{K_0}\rangle$. Suppose $N_b \sqsubset \{h\}_{K_0} \wedge v_0 \not\sqsubset \{h\}_{K_0}$. Since $N_b \neq \{h\}_{K_0}$, $N_b \sqsubset h$. Moreover, $v_0 \not\sqsubset h$, so the positive node is not minimal in S.

D. The trace $\text{tr}(p)$ has the form $\langle -K_0^{-1},\ -\{h\}_{K_0},\ +h\rangle$. If the positive node is minimal in S, then $v_0 \not\sqsubset h$ but $v_0 \sqsubset \{h\}_{K_0}$. Hence (using the assumption of free encryption) $h = N_a\, N_b\, B$ and $K_0 = K_A$. Thus, there exists a node m (the first on this strand) with $\text{term}(m) = K_A^{-1}$. Since by assumption, $K_A^{-1} \notin \mathbf{K}_\mathcal{P}$, we may apply Proposition 3.3 to infer that K_A^{-1} originates on a regular node. However, no initiator strand or responder strand originates K_A^{-1}.

S. The trace $\text{tr}(p)$ has the form $\langle -g\, h,\ +g,\ +h\rangle$. Assume $\text{term}(n_2) = g$; there is a symmetrical case if $\text{term}(n_2) = h$. Because $n_2 \in S$, $N_b \sqsubset g$ and $v_0 \not\sqsubset g$. (Note: by the minimality of n_2, we must have $v_0 \sqsubset g\, h$, so $v_0 \sqsubset h$, as v_0 is an encrypted value, not a concatenated value.)

Let $T = \{m \in \mathcal{C} : m \prec n_2 \wedge g\, h \sqsubset \text{term}(m)\}$. Every member of T is a penetrator node, because no regular node contains a subterm $g\, h$ where h contains any encrypted subterm.

T is non-empty because $\langle p, 1\rangle \in T$. Hence T has a minimal member m by Lemma 2.6, which is of positive sign by Lemma 2.7. Let us consider what kind of strand m can lie on.

M, F, T, K. Clearly a minimal member of T cannot lie on these strands.

S. If $g\,h \sqsubset \text{term}(m)$, where m is a positive node on a strand p' of kind **S**, then $g\,h \sqsubset \text{term}(\langle p', 1\rangle)$. Moreover, $\langle p', 1\rangle \prec m$, contradicting the minimality of m in T.

E. If $g\,h \sqsubset \text{term}(m)$, where m is a positive node on a strand p' of kind **E**, then $g\,h \sqsubset \text{term}(\langle p', 2\rangle)$. Moreover, $\langle p', 2\rangle \prec m$, contradicting the minimality of m in T.

D. If $g\,h \sqsubset \text{term}(m)$, where m is a positive node on a strand p' of kind **D**, then $g\,h \sqsubset \text{term}(\langle p', 2\rangle)$. Moreover, $\langle p', 2\rangle \prec m$, contradicting the minimality of m in T.

C. Suppose $g\,h \sqsubset \text{term}(m)$, where m is a positive node on a strand p' of kind **C**, and m is minimal in T. Then $g\,h = \text{term}(m)$, and p' has trace $\langle -g,\ -h,\ +g\,h\rangle$. Hence, $\text{term}(\langle p', 1\rangle) = \text{term}(n_2)$ and $\langle p', 1\rangle \prec n_2$, contradicting the minimality of n_2 in S.

Therefore n_2 does not lie on a penetrator strand, but must lie on a regular strand instead. ■

Definition 4.5 *Let n_2 be $\preceq$-minimal in $S = \{n \in \mathcal{C} : N_b \sqsubset \textit{term}(n) \wedge v_0 \not\sqsubset \textit{term}(n)\}$, and therefore regular and of positive sign.*

We show next that the strand containing n_2 also has a node in which v_0 $(= \{N_a\,N_b\,B\}_{K_A})$ occurs. This lemma is illustrated in Figure 2.

Lemma 4.6 *A node n_1 precedes n_2 on the same regular strand t, and $\textit{term}(n_1) = \{N_a\,N_b\,B\}_{K_A}$.*

PROOF. N_b originates at n_0 (Lemma 4.3), and originates uniquely in Σ (Assumption 3). Moreover, $n_2 \neq n_0$, because $v_0 \sqsubset \text{term}(n_0)$ while $v_0 \not\sqsubset \text{term}(n_2)$. Hence, N_b does not originate at n_2. So there is a node n_1 preceding n_2 on the same strand such that $N_b \sqsubset \text{term}(n_1)$. By the minimality property of n_2, $\{N_a\,N_b\,B\}_{K_A} \sqsubset \text{term}(n_1)$. However, as no regular node contains an encrypted term as a proper subterm, $\{N_a\,N_b\,B\}_{K_A} = \text{term}(n_1)$. ■

Lemma 4.7 *The regular strand t containing n_1 and n_2 is an initiator strand, and is contained in $\mathcal{C}$.*

PROOF. Node n_2 is a positive regular node and comes after a node (namely n_1) of the form $\{x\,y\,z\}_K$. Hence t is an initiator strand; if it were a responder strand, it would contain only a negative node after one of that form. Thus, n_1 and n_2 are the second and third nodes of t respectively. Since the last node of t is contained in $\mathcal{C}$, all previous nodes are also. ■

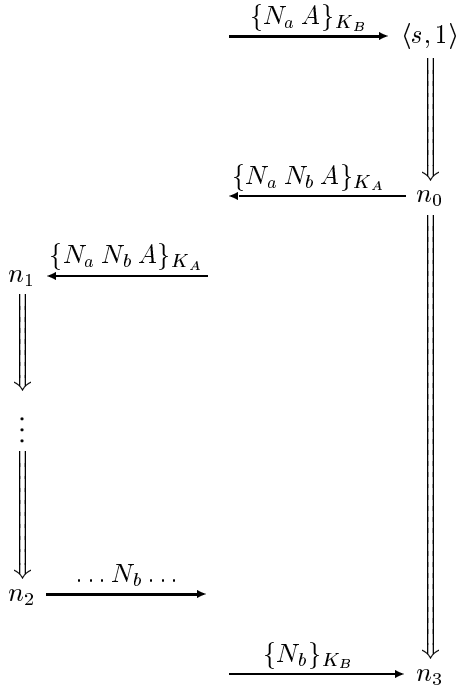

Figure 2. Node n_1 Contains v_0

PROOF OF PROPOSITION 4.2. Proposition 4.2 now follows immediately from Lemmas 4.6 and 4.7. ■

We have now proved the non-injective agreement property for the NSL responder. Injectivity follows easily on the assumption that the initiator chooses his value N_a so that it uniquely originates. If N_a is not uniquely originating, then the injectivity property is clearly false.

Proposition 4.8 *If Σ is an NSL space, $\mathcal{C}$ is a bundle, and N_a is uniquely originating in Σ, then there is at most one strand t with trace $\text{Init}[A, B, N_a, N_b]$ for any A, B, and N_b.*

PROOF. If any strand t has trace $\text{Init}[A, B, N_a, N_b]$ for any A, B, and N_b, then $\langle t, 1\rangle$ is positive, $N_a \sqsubset \text{term}\langle t, 1\rangle$, and N_a cannot possibly occur earlier on t. So N_a originates at node $\langle t, 1\rangle$. Hence, if N_a originates uniquely in Σ, there can be at most one such t. ■

The requirement that N_a and N_b be distinct is a peculiarity of our approach. Without this assumption, the theorem is false. The responder strand

$$\langle -\{N_a\,A\}_{K_B},\quad +\{N_a\,N_a\,B\}_{K_A},\quad -\{N_a\}_{K_B}\rangle$$

can be embedded in a bundle $\mathcal{C}$ in which N_a and A originate on **M**-nodes, and the final term $\{N_a\}_{K_B}$ is generated by the penetrator on the "off chance" that B will reuse the given nonce N_a. The responder's nonce N_b $(= N_a)$ does originate uniquely then; however, not on the responder's strand, but on an **M**-strand.

In a probabilistic model, we would assume that the choice of N_b is independent of the value of N_a. In this case, the penetrator's strategy will succeed sometimes, but no more frequently than randomly generating the bits to encrypt to make up the last message. Hence, this strategy may be safely ignored.

Thus, our strand space model can be more stringent than a faithful probabilistic model. An implementor can justify "cutting corners," for instance by not programming the check for $N_b = N_a$, by showing in the probabilistic model that an exploitation strategy has negligible probability of success, despite existing in the strand space model.

4.3 The Original Needham-Schroeder Protocol

This analysis also sheds light on why the original Needham-Schroeder protocol would be vulnerable. The analysis is exactly parallel except that the Lemma corresponding to Lemma 4.6 would read:

Lemma 4.9 *In the original Needham-Schroeder protocol, a node n_1 precedes n_2 on the same regular strand t, and term$(n_1) = \{N_a\ N_b\}_{K_A}$.*

With this weaker information, we can not conclude that t has a trace of the form Init$[A, B, N_a, N_b]$, because the responder's identity is not determined by the term $\{N_a\ N_b\}_{K_A}$, which is all that we know s and t agree on. We can only infer that t has trace Init$[A, C, N_a, N_b]$ for some C. This is exactly the weakness that Lowe's attack exploits.

4.4 Secrecy: The Responder's Nonce

We may use the same methods to show that the responder's nonce N_b remains secret in the protocol. For this result, we also need to assume that the responder's private key is not compromised. If it were, the penetrator could read N_b directly from the last message of the exchange.

Proposition 4.10 *Suppose:*

1. *Σ is an NSL space, and $\mathcal{C}$ is a bundle containing a responder's strand s with trace* Resp$[A, B, N_a, N_b]$*;*
2. *$K_A^{-1} \notin \mathcal{K}_\mathcal{P}$ and $K_B^{-1} \notin \mathcal{K}_\mathcal{P}$; and*
3. *$N_a \neq N_b$ and N_b is uniquely originating in Σ.*

Then for all nodes $m \in \mathcal{C}$ such that $N_b \sqsubset$ term(m), either $\{N_a\ N_b\ B\}_{K_A} \sqsubset$ term(m) or $\{N_b\}_{K_B} \sqsubset$ term(m). In particular, $N_b \neq$ term(m).

PROOF. Let $\Sigma, \mathcal{C}$, s, A, B, N_a, and N_b satisfy the hypotheses, and, as in Proposition 4.2, we will again refer to $\langle s, 2\rangle$ as n_0, and to its term $\{N_a\ N_b\ B\}_{K_A}$ as v_0. The node $\langle s, 3\rangle$ receives the value $\{N_b\}_{K_B}$; we will refer to this node as n_3 and its term as v_3. Consider the set:

$$S = \{n \in \mathcal{C} \quad : \quad N_b \sqsubset \text{term}(n) \\ \wedge \quad v_0 \not\sqsubset \text{term}(n) \ \wedge \ v_3 \not\sqsubset \text{term}(n)\}$$

If S is non-empty, then it has at least one $\preceq$-minimal element. We show first (Lemma 4.11) that such nodes are not regular. We next show (Lemma 4.12) that they are not penetrator nodes. Therefore S is empty, and the theorem holds.

Lemma 4.11 *No minimal member of S is a regular node.*

PROOF. Suppose instead that $m \in S$ is minimal and a regular node. The sign of m is positive by Lemma 2.7.

Node m cannot lie on s: Only n_0 is positive, and $v_0 =$ term(n_0), so n_0 is not in S.

Nor can m lie on a responder's strand $s' \neq s$. In that case, $m = \langle s', 2\rangle$, so term$(m) = \{N, N', C\}_{K_D}$. Since $N_b \sqsubset$ term(m), either $N_b = N$ or $N_b = N'$.

- If $N_b = N$, $N_b \sqsubset$ term$(\langle s', 1\rangle)$, because the first node $\langle s', 1\rangle$ is $\{N, D\}_{K_C} = \{N_b, D\}_{K_C}$. Moreover, $v_0 \not\sqsubset \{N_b, D\}_{K_C}$ and $v_3 \not\sqsubset \{N_b, D\}_{K_C}$. Hence $\langle s', 1\rangle \in S$. Since $\langle s', 1\rangle \prec m$, this contradicts the minimality of m.
- If $N_b \neq N$ and $N_b = N'$, then N_b originates at m, contradicting the assumption that N_b originates uniquely on n_0.

Suppose next that m lies on an initiator strand s'. It must be either the first or third node.

- If $m = \langle s', 1\rangle$, then since $N_b \sqsubset$ term(m), N_b originates at m, contradicting the assumption that N_b originates uniquely on n_0.
- If $m = \langle s', 3\rangle$, then term$(m) = \{N_b\}_{K_C}$. So the second node $\langle s', 2\rangle$ is of the form $\{x\ N_b\ C\}_K$. However, $C \neq B$, because otherwise $v_3 =$ term(m). Hence $\langle s', 2\rangle \prec m$ is in S, contradicting the minimality of m. ■

Lemma 4.12 *No minimal member of S is a penetrator node.*

PROOF SKETCH. The proof is almost identical to the proof of Lemma 4.4. The only significant difference is that when the penetrator strand is of type **D**, we must consider two cases. In one case, $h = N_a\ N_b\ B$ and $K_0 = K_A$, which are the plaintext and key that produce v_0. In the other case, $h = N_b$ and $K_0 = K_B$, which are the plaintext and key that produce v_3. Hence, we must apply Proposition 3.3 to each of the two private keys, which explains the need to assume both uncompromised. ■

4.5 The Initiator's Guarantees: Secrecy and Agreement

The proof of the secrecy of the initiator's nonce N_a is very similar to the proof we have just given.

Proposition 4.13 *Suppose:*

1. Σ *is an NSL space, and* $\mathcal{C}$ *is a bundle containing an initiator's strand* s *with trace* $\text{Init}[A, B, N_a, N_b]$*;*
2. $K_A^{-1} \notin \mathcal{K}_\mathcal{P}$ *and* $K_B^{-1} \notin \mathcal{K}_\mathcal{P}$*; and*
3. N_a *is uniquely originating in* Σ*.*

Then for all nodes $m \in \mathcal{C}$ *such that* $N_a \sqsubset term(m)$*, either* $\{N_a\, A\}_{K_B} \sqsubset term(m)$ *or* $\{N_a\, N_b\, B\}_{K_A} \sqsubset term(m)$*. In particular,* $N_a \neq term(m)$*.*

By contrast, the initiator's guarantee of agreement is essentially different. In particular, it requires a stronger hypothesis than Proposition 4.2, namely that both private keys K_A^{-1} and K_B^{-1} are uncompromised. Not surprisingly, if $K_B^{-1} \in \mathbf{K}_\mathcal{P}$, then the penetrator can complete the entire exchange with no activity on B's part.

Somewhat more surprising is this: If $K_A^{-1} \in \mathbf{K}_\mathcal{P}$, then the penetrator can read B's reply $\{N_a\, N_b\, B\}_{K_A}$, substituting a different reply $\{N_a\, N'\, B\}_{K_A}$. This attack prevents us from proving agreement for the initiator assuming only that the responder's private key is uncompromised. Indeed, a proof approach based on an analogy with Proposition 4.2 fails.

However, we can prove an agreement theorem using the secrecy of N_a as a lemma.

Proposition 4.14 *Suppose:*

1. Σ *is an NSL space and* $\mathcal{C}$ *is a bundle containing an initiator's strand* s *with trace* $\text{Init}[A, B, N_a, N_b]$*;*
2. $K_A^{-1} \notin \mathcal{K}_\mathcal{P}$ *and* $K_B^{-1} \notin \mathcal{K}_\mathcal{P}$*; and*
3. N_a *is uniquely originating in* Σ*.*

Then $\mathcal{C}$ *contains the first two nodes of a responder's strand* t *with trace* $\text{Resp}[A, B, N_a, N_b]$*.*

PROOF SKETCH. Consider the set $\{m \in \mathcal{C} : \{N_a\, N_b\, B\}_{K_A} \sqsubset \text{term}(m)\}$. It is non-empty because it contains $\langle s, 2\rangle$. So it contains a minimal member m_0. If m_0 lies on a regular strand t, then t can be shown to have trace $\text{Resp}[A, B, N_a, N_b]$, and to have two nodes (at least) in $\mathcal{C}$.

If instead m_0 lies on a penetrator strand t, then t can be shown to be an **E**-strand with trace

$$\langle -K_A, \quad -N_a\, N_b\, B, \quad +\{N_a\, N_b\, B\}_{K_A} \rangle$$

But this contradicts Proposition 4.13, which implies that N_a does not appear in the form shown in node $\langle t, 2\rangle$.

5 Discussion

In this paper, we have developed a new framework for proving the correctness of cryptographic protocols, and we have applied it to the Needham-Schroeder-Lowe protocol.

The framework allows us to use mathematically straightforward methods to justify protocols. These methods primarily exploit two partial orderings, namely the subterm relation $\sqsubset$ between terms and the $\preceq$ relation between nodes. Inductive characteristics of the $\preceq$ ordering are proved via a least element principle. Inductive characteristics of the $\sqsubset$ relation can also be exploited [25, 26].

Proofs carried out in the strand space framework turn on detailed protocol behavior, and therefore appear more reliable than more "conceptual" proofs such as proofs in belief logics [3, 9]. Moreover, the proofs are intuitive enough that mere mortals can carry them out correctly without the need for mechanized support.

In each of the examples we have studied, as documented in [25], we have discovered new information about the conditions under which the protocol is correct. We have found that:

- The responder's agreement guarantee in the Needham-Schroeder-Lowe protocol holds even if the responder's private key has been compromised. By contrast, the initiator's agreement guarantee presupposes that neither the initiator nor the responder has had his private key compromised (Section 4.5).
- In the Otway-Rees protocol, even if both the responder and the initiator receive keys, they may receive different keys. This is essentially due to Otway-Rees establishing a non-injective sort of agreement between each principal and the server.
- In the Yahalom protocol, if there are multiple trusted servers, participants may play the role of a server as well as the role of an ordinary participant, so long as a particular symmetry is avoided. Otherwise attacks are possible.

Thus, the strand space approach leads to a precise characterization of the validity of the protocols.

Our work is closely related to Paulson's inductive approach [21, 20, 22]. Paulson models a protocol as a set of rules for extending a sequence of events; some of these rules represent actions by legitimate participants, while others represent actions by the penetrator. A sequence of events generated by these rules corresponds roughly to a bundle. Paulson expresses authentication goals and secrecy goals as properties of these sequences, which he can then prove by induction on the way that the sequence is generated. The general-purpose theorem-proving system Isabelle [19] provides mechanical support for the reasoning.

By contrast, our approach uses a partially ordered structure, the bundle. As we mentioned, Lemma 2.6 is in effect an induction principle on the partial order $\preceq_{\mathcal{C}}$. The nodes in the bundle are organized into strands. Naturally, every bundle may be linearized into an event sequence in at least one way, while any event sequence determines a bundle.

However, we think there are two advantages to our approach. First, the bundle contains exactly the causally relevant information. There is no ordering relation between two nodes unless the causality determined by the basic relations $\rightarrow$ and $\Rightarrow$ requires one, and this simplifies inductive arguments. Second, the strand captures a great deal of information. A particular strand may be known to have nodes in a bundle (e.g. because a value originates uniquely on it). From this we can identify the whole sequence of relevant actions for that participant, which aids in isolating the exact agreement properties the protocol satisfies. We believe this is why our results are somewhat sharper than others in the literature.

The strand space framework can also be used in other ways, apart from being used simply to prove a protocol correct. For instance, it could be used to give an alternate semantics for belief logics, whether applied to cryptographic protocols [3, 2] or distributed systems more broadly [10], in contrast to the more usual semantical approaches based on sequences of events or states. The localization that the notion of strand offers should help to refine and sharpen such models. Alternatively, results about authentication protocols proved in a strand space context can be imported into the more usual linear models by linearizing the bundles.

Acknowledgements. We are grateful to Sylvan Pinsky, Al Maneki, and their colleagues at NSA for support, encouragement, and discussions. Shim Berkovits, Marion Michaud, and John Vasak patiently helped us improve the presentation. Peter Ryan taught us a great deal about the field, and provided the initial impetus for doing the work. The anonymous referees made shrewd and useful comments.

References

[1] M. Abadi and A. D. Gordon. Reasoning about cryptographic protocols in the spi calculus. In *CONCUR 97*, Lecture Notes in Computer Science, pages 59–73. Springer-Verlag, July 1997.

[2] M. Abadi and M. R. Tuttle. A semantics for a logic of authentication. In *Proceedings of the 10th ACM Symposium on Principles of Distributed Computing*, pages 201–216, August 1991.

[3] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *Proceedings of the Royal Society*, Series A, 426(1871):233–271, December 1989. Also appeared as SRC Research Report 39 and, in a shortened form, in ACM Transactions on Computer Systems 8, 1 (February 1990), 18-36.

[4] U. Carlsen. Cryptographic protocol flaws. In *Proceedings 7th IEEE Computer Security Foundations Workshop*, pages 192–200. IEEE Computer Society, 1994.

[5] J. Clark and J. Jacob. On the security of recent protocols. *Information Processing Letters*, 56(3):151–155, Nov. 1995.

[6] D. Denning and G. Sacco. Timestamps in key distribution protocols. *Communications of the ACM*, 24(8), Aug. 1981.

[7] D. Dolev and A. Yao. On the security of public-key protocols. *IEEE Transactions on Information Theory*, 29:198–208, 1983.

[8] R. Focardi and R. Gorrieri. The compositional security checker: A tool for the verification of information flow security properties. *IEEE Transactions on Software Engineering*, 23(9), September 1997.

[9] L. Gong, R. Needham, and R. Yahalom. Reasoning about belief in cryptographic protocols. In D. Cooper and T. Lunt, editors, *Proceedings 1990 IEEE Symposium on Research in Security and Privacy*, pages 234–248. IEEE Computer Society, May 1990.

[10] J. Y. Halpern. Reasoning about knowledge: A survey. In D. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 4, pages 1–34. Oxford University Press, 1995.

[11] G. Lowe. An attack on the Needham-Schroeder public key authentication protocol. *Information Processing Letters*, 56(3):131–136, Nov. 1995.

[12] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Proceeedings of* TACAS, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166. Springer Verlag, 1996.

[13] G. Lowe. Casper: A compiler for the analysis of security protocols. In *10th Computer Security Foundations Workshop Proceedings*, pages 18–30. IEEE Computer Society Press, 1997.

[14] G. Lowe. A heirarchy of authentication specifications. In *10th Computer Security Foundations Workshop Proceedings*, pages 31–43. IEEE Computer Society Press, 1997.

[15] W. Marrero, E. Clarke, and S. Jha. A model checker for authentication protocols. In C. Meadows and H. Orman, editors, *Proceedings of the DIMACS Workshop on Design and Verification of Security Protocols*. DIMACS, Rutgers University, September 1997.

[16] J. H. Moore. Protocol failures in cryptosystems. *Proceedings of the IEEE*, 76(5), May 1988.

[17] R. Needham and M. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12), Dec. 1978.

[18] S. Patel. Number theoretic attacks on secure password schemes. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, pages 236–247. IEEE Computer Society Press, May 1997.

[19] L. C. Paulson. *Isabelle: A Generic Theorem Prover*. Number 828 in LNCS. Springer, 1994.

[20] L. C. Paulson. Mechanized proofs of a recursive authentication protocol. In *10th IEEE Computer Security Foundations Workshop*, pages 84–94. IEEE Computer Society Press, 1997.

[21] L. C. Paulson. Proving properties of security protocols by induction. In *10th IEEE Computer Security Foundations Workshop*, pages 70–83. IEEE Computer Society Press, 1997.

[22] L. C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 1998. Also Report 443, Cambridge University Computer Lab.

[23] A. W. Roscoe. Intensional specifications of security protocols. In *Proceedings of the 9th IEEE Computer Security Foundations Workshop*, pages 28–38, 1996.

[24] S. Schneider. Verifying authentication protocols with CSP. In *Proceedings of the 10th IEEE Computer Security Foundations Workshop*, pages 3–17. IEEE Computer Society Press, 1997.

[25] F. J. Thayer Fábrega, J. C. Herzog, and J. D. Guttman. Strand spaces. Technical report, The MITRE Corporation, November 1997.

[26] F. J. Thayer Fábrega, J. C. Herzog, and J. D. Guttman. Honest ideals on strand spaces. Submitted for publication, February 1998.

[27] T. Y. C. Woo and S. S. Lam. Verifying authentication protocols: Methodology and example. In *Proc. Int. Conference on Network Protocols*, October 1993.